基于数据密度的半监督自训练分类算法 *

艾震鹏，王振友

(广东工业大学 应用数学学院，广州 510520)

摘　要：在实际的分类任务中，无标记样本数量充足而有标记样本数量稀少的情况经常出现，目前处理这种情况的常用方法是半监督自训练分类算法。提出了一种基于数据密度的半监督自训练分类算法，该算法首先依据数据的密度对数据集进行划分，从而确定数据的空间结构；然后再按照数据的空间结构对分类器进行自训练的迭代，最终得到一个新的分类器。在 UCI 中 6 个数据集上的实验结果表明，与三种监督学习算法以及其分别对应的自训练版本相比，提出的算法分类效果更好。

关键词：半监督学习；自训练；密度；分类

中图分类号：TP301.6　　doi: 10.3969/j.issn.1001-3695.2017.12.0753

Self-training semi-supervised classification based on density of data

Ai Zhenpeng, Wang Zhenyou

(*Department of Applied Mathematicslege Guangdong University of Technology*, *Guangzhou* 510520, *China*)

Abstract: It is a common problem in many practical applications that unlabeled samples is sufficient but labeled ones is very rare. A successful method to tackle this problem is self-training semi-supervised classification. In this paper, a self-training semi-supervised classification method is introduced, in which entire data is divided into three parts based on density of data, so that the real structure of data space could be found. And then, a framework for self-training semi-supervised classification, in which the structure of data space is integrated into the self-training iterative process to help train a better classifier, is proposed. Experiments on 6 data sets from UCI show that the classifier get from the method proposed have a better performance than the ones get from supervised method with few labeled samples and standard self-training semi-supervised classification method.

Key words: semi-supervised; self-training; density; classification

0　引言

在数据挖掘和机器学习中，监督学习是一个活跃的研究领域，到目前为止，已经被广泛应用于异常检测，生物医学，人脸识别和文本分类等诸多地方[1−4]。监督学习依赖于利用充足的有标记样本训练出一个优秀的分类器，再利用该分类器来完成分类任务。然而在实际应用中，获得充足的有标记样本本身就是一件困难的事情，与此相反，获得大量的无标记样本则会相对容易很多。而监督学习在只有少量的有标记样本时往往不能训练出一个性能良好的分类器，并且因为没有对大量的无标记样本加以利用，造成了极大的浪费。半监督学习的目的就是要使用无标记样本来提升监督学习算法的性能。

近年来，有许多半监督学习的方法被提出。这些方法大多基于数据的聚类假设和流形假设。例如 Joachims 提出的 TSVM 算法[5]就是基于聚类假设。而 Zhu 等人提出的基于图的半监督学习[6]则是利用了流形假设。还有一类半监督学习的方法是自标记的方法，即自训练和协同训练。其中自训练方法首先利用初始的有标记样本训练出一个分类器，然后利用该分类器对无标记样本进行打标记，并选择置信度高的被打上标记的样本加入到有标记样本集中，不断迭代，直到达到停止条件，以这样的方式来扩大有标记样本的容量，例如 Yarowsky 的文章中提到的自训练算法[7]。

然而，当初始有标记样本不能够体现整个数据的空间结构，或者有标记样本数量过少时，自训练算法的效果很不理想。因为最初始的分类器对无标记样本的分类效果并不好，那么之后的打标记过程就相当于引入了噪声，在这样的情况下，自训练算法可能会比单纯地使用有标记样本进行监督学习的效果更差。Adankon 和 Cheriet 提出了一种叫 help-training 的自训练算法[8]，使用生成式模型来训练分类器，但这个方法依然没有解决自训练的问题，因为生成式的模型仅仅是依靠有标记样本得到的。

收稿日期：2017-12-03；**修回日期：**2018-01-16　**基金项目：**广州市科技计划资助项目（201707010435）；广东省研究生教育创新改革项目（2014JGXM-MS17）

作者简介：艾震鹏（1993-），男，四川简阳人，硕士研究生，主要研究方向为数据挖掘、机器学习（15876501504@163.com）；王振友（1979-），男，副教授，博士，主要研究方向为机器学习、最优化理论与方法、计算生物学.

Gan 等人引入了模糊 K-means 来优化自训练算法[9]，并取得了较好的效果，但是对于非高斯分布的数据，分类效果并不理想。在本文中，本文提出了一种基于数据密度的自训练分类算法，算法首先使用基于密度的方法确定整个数据集的真实空间结构，然后按照数据的空间结构迭代地对分类器进行训练。本文提出的算法有以下两点优势：a)对非高斯分布的数据依然有较好的分类效果。值得一提的是，在实际应用中，数据许多情况下并不是高斯分布;b)可以使用任意监督学习的算法作为分类器，充分发挥不同监督学习算法的优势。

1　数据空间结构的确定

在处理无标记数据时，聚类是一种能够发现数据空间结构的重要方法。聚类算法有很多，比如 k-means 就是一个易于实现并在多数情况下效果良好的算法，但 k-means 的缺点在于处理非高斯分布的数据时效果不佳。而 DBSCAN 是一个比较有代表性的基于密度的聚类算法[10]，较 k-means 的优势在于它将簇定义为密度相连的点的最大集合，能够把具有足够高密度的区域划分为簇，并可在有噪声的数据空间中发现任意空间结构的聚类。本文对数据空间结构的确定方法就是基于这样的想法。

1.1　基于数据密度的空间结构确定方法

在本文中，设 $L=\{(x_i,y_i)\}$ 是有标记样本集，其中 x_i 是训练样本，y_i 是它的标记，$y_i\in\{\omega_1,\omega_2,\cdots,\omega_s\}$，$i=1,2,\cdots,n$，$s$ 是类别数。$U=\{x_{n+1},x_{n+2},\cdots,x_m\}$ 是无标记样本集。则样本点 x_i 的密度定义如下：

$$\rho_i=\sum_j\delta(d_{ij}-d_c) \tag{1}$$

其中:

$$\delta(x)=\begin{cases}1, x<0\\0, x\geq 0\end{cases}$$

d_{ij} 表示 x_i 到 x_j 的距离，d_c 被称为截断距离，是一个与数据集本身有关的常数。显然，ρ_i 的值等于到 x_i 距离小于 d_c 的点的个数。取密度阈值为 ρ_0，对于 x_i 来说，若 ρ_i 大于 ρ_0，则 x_i 就被称为核心点。若 ρ_i 小于 ρ_0，且 x_i 的 d_c 领域内有核心点，则 x_i 就被称为边界点，若 x_i 既不是核心点又不是边界点，那么 x_i 就叫做噪声点。通过将数据集中的每一个样本点标注为核心点、边界点或噪声点，那么不论该数据是高斯分布还是其他分布，本文都能够发现数据真实的空间结构。在图 1 中，A、B 两类总共 40 个样本点 $\{x_1,x_2,\cdots,x_{40}\}$ 按照如下分布分别随机产生：

A 类：$\{(a,b)\mid a\sim N(3.5,0.5), b\sim U(1,9)\}, i=1,2,\cdots,20$

B 类：$\{(a,b)\mid a\sim N(6.5,0.5), b\sim U(1,9)\}, i=21,22,\cdots,40$

其中 $N(\mu,\sigma^2)$ 是以 μ 为均值，σ^2 为方差的高斯分布，$U(a,b)$ 是从 a 到 b 的均匀分布。接下来利用式(1)，首先可以算出每个点的密度值 ρ_i，然后按照上述方法将样本点分为核心点，边界点和噪声点，最后如图 2 所示，得到了数据的真实空间结构。

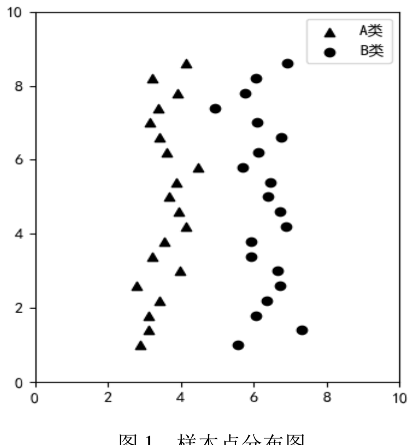


图 1　样本点分布图

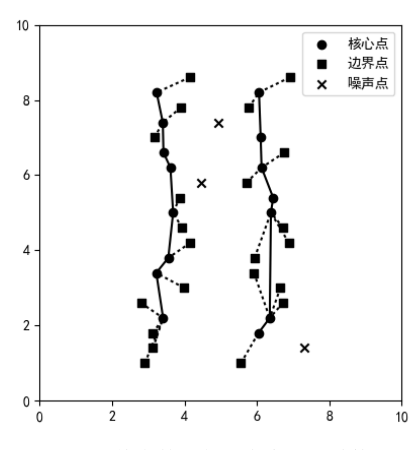


图 2　整个数据集的真实空间结构

1.2　参数 d_c 和 ρ_0 的选取

根据文献[11]中的结论，d_c 的取值要使得所有样本点的平均邻居个数在样本总数的 1%到 2%范围内。具体选取方式如下：

首先计算出所有的 d_{ij}，然后将 d_{ij} $(i<j)$ 按照升序排列，得到序列 $d_1,d_2,\cdots,d_M$，其中 $d_1<d_2<\cdots<d_M$，$M=\frac{1}{2}N(N-1)$，N 是样本点数目，最后令

$$d_c=d_{f(Mt)} \tag{2}$$

$t\in(0,1)$,用于将 d_c 控制在合适的范围内，$f(Mt)$ 表示对 Mt 的乘积进行四舍五入取整。

而 ρ_0 的选取则以核心点能够描述出数据的整体空间结构为目标，局部更为细微的结构则由依赖于核心点和 d_c 的边界点来完成。ρ_0 的具体选取方式与 d_c 类似：

首先计算出所有的 ρ_i，并将 ρ_i 按照升序排列，得到 $\rho_1,\rho_2,\cdots,\rho_N$，然后令

$$\rho_0=\rho_{g(Nh)} \tag{3}$$

与式(2)相似，$g(Nh)$ 表示对 Nh 的乘积四舍五入取整，而 $h\in(0,1)$ 用于将 ρ_0 控制在合适范围内。

如前文所述，不论数据是高斯分布还是其他形状的分布类型，只需通过式②③取到合适的阈值 ρ_0 和 d_c，并利用式①计算每个样本点 x_i 的密度 ρ_i，就可以将数据分为核心点，边界点和噪声点，进而得到数据的空间结构，而没有迭代的过程，因此

计算速度很快，非常适合用在自训练算法中。

2 基于数据空间结构的自训练算法

在本文的这个部分将会给出一个半监督自训练的具体方法，在该方法中，数据集的真实空间结构将会在迭代训练分类器时被考虑进去。具体步骤描述如下：

a)通过将所有样本点分类为核心点，边界点和噪声点，得到数据集的真实空间结构

b)利用初始有标记集 L 训练得到一个分类器 C

c)利用分类器 C 对 U 中所有核心点迭代地打标记，同时更新 L 和 U ，并再次利用 L 训练 C，直到 U 中所有核心点都被打上标记

d)与 c)类似，利用分类器 C 对 U 中所有边界点迭代地打标记，同时更新 L 和 U ，并再次利用 L 训练 C ,直到 U 中所有边界点都被打上标记。

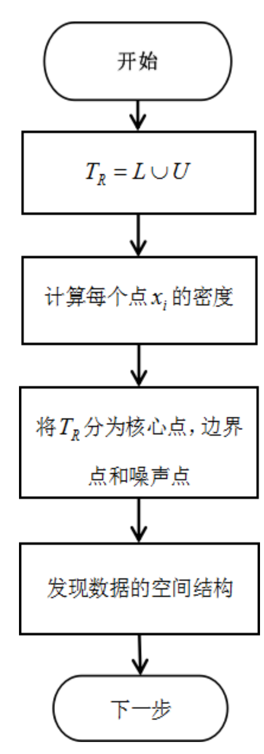


图 3 确定数据空间结构

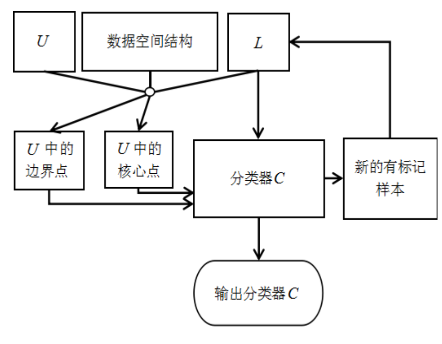


图 4 自训练过程

如图 3 是算法 a)步中确定数据空间结构的流程图，图 4 是 b)步~d)步中对 U 中核心点和边界点迭代打标记，并得到最终分类器 C 的流程图。算法伪代码如下：

输入：有标记样本集 L，无标记样本集 U，参数 ρ_0， d_c。

输出：分类器 C。

```
for T_R = L ∪ U 中的每个点 x_i DO
  利用式（1）计算 ρ_i
  if ρ_i > ρ_0 then
    将 x_i 标记为核心点
  else if ρ_i < ρ_0 and 邻域 δ(x_i, d_c) 中有核心点 then
    将 x_i 标记为边界点
  else
    将 x_i 标记为噪声点
  do
    利用 L 训练得到分类器 C
  while U 中有核心点 DO
  取核心点 x_c ∈ U，分类器 C 对 x_c 打标记后得到新的有标记样本 x_c'；更新 L = L ∪ x_c'，U = U / x_c，并用 L 训练分类器 C
  while U 中有边界点 do
    取边界点 x_b ∈ U，分类器 C 对 x_b 打标记后得到新的有标记样本 x_b'；更新 L = L ∪ x_b'，U = U / x_b，并用 L 训练分类器 C
return 分类器 C
```

3 实验结果与分析

在本章中，将基于数据密度的自训练算法与三种监督学习算法以及它们分别对应的标准自训练算法进行比较，并对得到的实验结果进行分析。

3.1 实验设计

3.1.1 实验数据

在实验中，使用了六个真实数据集来测试算法的表现，这些数据集全部来自于 UCI，表 1 中是这些数据集的详细信息。在实验中，本文首先将每一个数据集随机划分为两部分，第一部分占总样本数的 25%，作为测试集 T 。剩下 75%作为训练集。然后再将训练集随机分为两部分，其中 10%保留标记，作为有标记样本集 L，90%去掉标记，作为无标记样本集 U 。

表 1 数据集信息

数据名称	样本数目	属性数目	类别数目
iris	150	4	3
banknote	1372	4	2
seeds	210	7	3
haberman	306	3	2
pima	768	8	2
wine	178	13	3

3.1.2 参数设置

选择了最近邻算法(KNN)、支持向量机(SVM)、单层神经网

络(SLNN)以及它们分别对应的标准自训练算法与本文提出的基于数据密度的自训练算法进行对比，这些算法的参数信息展示在表 2 中。表 3 中的是将数据标准化处理以后，不同数据集 ρ_0 和 d_c 的选取情况。

表 2　算法相关参数

算法名称	记号	参数
最近邻算法	KNN	最近邻数 K=5
支持向量机	SVM	-
单层神经网络	SLNN	隐藏层神经元数目 64，最大迭代数 2000，学习率 0.005
标准自训练算法	ST	-
改进的自训练算法	OST	-

表 3　ρ_0 和 d_c 选取情况

数据名称	ρ_0	d_c
iris	3	0.3877
banknote	20	0.4895
seeds	4	0.7842
haberman	6	3.0000
pima	8	1.3339
wine	2	1.9763

3.2　实验结果

表 4~6 中是 50 次实验结果的平均正确率和方差，每一次实验都重新随机抽样得到 T 、 L 和 U 。

表 4　基于 KNN 的实验结果

数据集	算法		
	KNN	ST-KNN	OST-KNN
iris	0.701/0.053	0.726/0.077	**0.743/0.036**
banknote	0.954/0.014	0.970/0.016	**0.979/0.009**
seeds	0.870/0.041	0.899/0.060	**0.905/0.025**
haberman	0.712/0.062	0.717/0.085	**0.738/0.040**
pima	0.665/0.047	0.689/0.066	**0.710/0.031**
wine	0.913/0.068	0.957/0.104	**0.969/0.047**

表 5　基于 SVM 的实验结果

数据集	算法		
	SVM	ST-SVM	OST-SVM
iris	0.734/0.042	0.744/0.054	**0.753/0.028**
banknote	0.972/0.009	**0.988**/0.018	0.979/**0.006**
seeds	0.913/0.037	**0.920**/0.051	0.914/**0.019**
haberman	0.720/0.051	0.747/0.076	**0.758/0.038**
pima	0.691/0.039	0.702/0.042	**0.720/0.025**
wine	0.966/0.041	0.969/0.042	**0.979/0.027**

表 6　基于 SLNN 的实验结果

数据集	算法		
	SLNN	ST-SLNN	OST-SLNN
iris	0.701/0.055	0.727/0.064	**0.736/0.039**
banknote	0.979/0.008	0.981/0.016	**0.988/0.006**
seeds	0.886/0.041	0.908/0.068	**0.914/0.030**
haberman	0.694/0.131	0.682/0.133	**0.733/0.062**
pima	0.683/0.045	0.710/0.062	**0.712/0.033**
wine	0.945/0.048	0.960/0.060	**0.979/0.034**

通过表 4~6 的结果可知，提出的半监督自训练算法在所选取六个样本集上，无论是从正确率还是从稳定性来说，全部优于仅仅使用有标记样本的监督学习算法，而在大多数样本上的表现比标准的半监督自训练算法要好，且明显更稳定。

4　结束语

本文提出了一种基于数据密度的半监督自训练分类算法，算法利用密度和距离将数据集划分成三类，进而确定数据的空间结构，然后按照数据的空间结构对分类器进行自训练的迭代，最终得到一个新的分类器。在本文中，选择了 UCI 上的 6 个真实数据集进行实验，并以三种常用的监督学习算法所得到的分类器为基础来进行自训练。实验结果表明，通过本文提出的算法得到的分类器，在分类效果上总体优于仅仅使用有标记样本进行监督学习和使用传统自训练算法得到的分类器。

在今后，将进一步对距离参数 d_c 和密度参数 ρ_0 的自适应取值方法进行研究，避免需要多次的实验来人为确定这两个参数。

参考文献：

[1] Ashfaq R A R, Wang Xizhao, Huang Joshua Zhexue, *et al.* Fuzziness based semi-supervised learning approach for intrusion detection system [J]. Information Sciences, 2017, 378 (C): 484-497.

[2] Xu Guangru, Zhang Minghui, Zhu Hongxing, *et al.* A 15-gene signature for prediction of colon cancer recurrence and prognosis based on SVM [J]. Gene, 2017, 604: 33-40.

[3] Vijayan A, Kareem S, Kizhakkethottam J J, *et al.* Face recognition across gender transformation using SVM classifier [J]. Procedia Technology, 2016, 24: 1366-1373.

[4] Pavlinek M, Podgorelec V. Text classification method based on self-training and LDA topic models [J]. Expert Systems with Applications, 2017, 80: 83-93.

[5] Joachims T. Transductive inference for text classification using support vector machines [C]// Proc of the 16th International Conference on Machine Learning. 1999: 200-209.

[6] Zhou D, Bousquet O, Lal T N, *et al.* Learning with local and global consistencyAdvances in Neural Information Processing Systems. Cambridge: MIT Press, 2004: 321-328.

[7] Yarowsky D. Unsupervised word sense disambiguation rivaling supervised methods [C]// Proc of the 33rd Annual Meeting of the Association for Computational Linguistics. 1995: 189-196.

[8] Adankon M M, Cheriet M. Help-training for semi-supervised support vector Machines [J]. Pattern Recognit, 2011, 44 (9): 2220-2230.

[9] Gan Haitao, Sang Nong, Huang Rui, *et al.* Using clustering analysis to improve semi-supervised classification [J]. Neurocomputing, 2013, 101: 290-298.

[10] Kumar K M, Reddy A R M. A fast DBSCAN clustering algorithm by accelerating neighbor searching using Groups method [J]. Pattern Recognition, 2016, 58: 39-48.

[11] Rodriguez A, Laio A. Clustering by fast search and find of density peaks [J]. Science, 2014, 344 (6191): 1492-1496.